(19) Japanese Patent Office
(11) Laid-open Patent Application
(12) Japanese Laid-open Patent Application Publication
No. Hei04-371989 (JP-A-4-371989)
(51)

Int. Cl.[5]		Classification Symbol	JPO Ref. No.
G09B	15/00	Z	6763-2C
	15/02		6763-2C
G10G	3/04	A	7346-5H

(43) Published: December 24, 1992
Request for Examination: No
Number of Claims: 6
(17 pages in total)

(54) Title of the Invention: MUSIC INFORMATION DISPLAY SYSTEM

(21) Application No.: Hei03-149119
(22) Applied: June 21, 1991

(71) Applicant: 000001410
Kawai Musical Instruments Mfg. Co., Ltd.
200 Terajima-cho, Hamamatsu-shi, Shizuoka
Japan
(72) Inventor: OKA, Masaaki
c/o Kawai Musical Instruments Mfg. Co., Ltd.
200 Terajima-cho, Hamamatsu-shi, Shizuoka
Japan
(74) Agent: WAKAHARA, Seiichi Patent Attorney

(57) [Abstract]
[Object]

A music information display system of the present invention sequentially searches displayed musical symbol information and arranges them in order from the beginning of a musical score to create music performance information. This allows displaying the music score while allowing one to listen to the performance based on the displayed music score, resulting in significant effectiveness in music composition and education.

[Constitution]

In a display code memory 131, a stave is first searched (Steps 31 through 46), and then musical symbols arranged within the stave are searched (Steps 51 through 59). Also, code data KD for the musical symbols (Steps 75, 61, 63, 65, 68, 70, 81, 72, 83, 85...) is converted into their respective music performance data MP (Steps 76 through 80, 62, 64, 66 through 67, 69, 82, 71 through 74, 84, 86...). Automatic music performance can be done based on the music performance data MP (Step 60). The display code memory 131 stores the code data KD for each musical symbol as display point coordinate data KC for a display entry point 24 in FIG. 2.

[0009]

[Embodiment]

1. Display Screen 20

FIG. 1 shows a display screen 20 of a CRT (cathode ray tube) 5. On the left of the display screen 20, symbol icons 22 of the musical symbols, such as a note (rest) and a stave, are displayed. Moving a cursor 25 of a mouse 1 onto a symbol icon 22 of a desired musical symbol and pressing a left button 1a of the mouse 1 allows selecting the desired musical symbol. After the selection, further moving the cursor 25 of the mouse 1 to a desired entry point within a viewport 21 and pressing the left click button 1a of the mouse 1 allows a display of a graphic pattern 26 of the musical symbol at this desired entry point. A liquid crystal display device or plasma display may be used in place of the aforementioned CRT 5.

[0010]

On the upper side of the display screen 20, command icons 23 for performance, sound, file and the like are displayed. Moving the cursor 25 of the mouse 1 onto a command icon 23 of a desired command and pressing the left button 1a of the mouse 1 allows selecting this desired command process. The viewport 21 occupies most of the display screen 20 and displays the graphic patterns 26 of the composed music score.

[0011]

FIG. 2 shows an example of the graphic patterns 26 of the music score displayed on the viewport 21. The viewport 21 is comprised of display parts 27 divided by a unit (3 bit length * 5 bit width). An entry of the graphic pattern 26 of each musical symbol is assigned to each display part 27.

[0012]

However, the graphic pattern 26 of each musical symbol is displayed across the plural display parts 27 as shown in FIG. 2. The graphic pattern 26 of each musical symbol has a display entry point 24 for displaying the entry point. The display part 27, to which the display entry point 24 belongs, is regarded as the entry point of the graphic pattern 26 for the musical

symbol.

[0013]

The musical symbol icons 22 are displayed as the graphic patterns 26. They indicate, from the top to bottom of the display screen, note (rest), stave, clef, concert bracket, key signature, bar line, accidental mark, repeat mark, time signature, tie (slur), crescendo (decrescendo), brace, dynamic mark, articulation, performance mark, tempo mark, beam, lyric words, tuplet and free graphic. The cursor 25 of the mouse 1 which turned to a musical symbol is moved within the viewport 21. The musical symbol changes in a cyclic manner each time clicking the right button 1b of the mouse 1.

[0014]

For example, the musical symbol indicating a note (rest) changes in a cyclic manner to upward quarter note, then downward quarter note, quarter rest, and again upward quarter note, … or changes to full note, then half note, quarter note, eighth note, sixteenth note, thirty-second note, and again full note… . In this case, pressing a numeric keypad "1" of a keyboard 3 allows one single dot to be added and pressing a numeric keypad "2" allows two dots to be added. The musical symbol indicating a stave changes to five-line type, then a single line type … in a cyclic manner. The musical symbol indicating a clef changes to G clef, then f clef, C clef, and again G clef … in a cyclic manner. The time signature is specified with two numbers, representing the numerator and denominator. These numbers individually change in a cyclic manner. In this case, clicking the right button 1b allows cyclic changes of the number of the numerator, after the number of the denominator is determined by clicking the left button 1a. The number of the numerator is stored as additional information of the time signature.

[0015]

The musical symbol indicating a accidental mark changes, for example, to sharp, then flat, natural, and again sharp … in a cyclic manner. The musical symbol indicating a key signature also changes in almost the same manner. The musical symbol

indicating a dynamic mark changes, for example, to fortissimo ff, forte f, …then pianissimo pp, and again fortissimo ff … in a cyclic manner. The dynamic mark may also include fortissississimo fff or louder one, pianississimo ppp or softer one, and diminuendo dim. The tempo mark is entered by pressing a numeric keypad on the keyboard 3. The entry of these musical symbols can be made by any method other than the aforementioned one, such as selecting each musical symbol displayed on the window, and entering them using the keyboard 3.

[0016]

For the musical symbols that require to specify the beginning and end, clicking the left button 1a of the mouse 1 at the beginning point and again clicking the same button at the end point allows a length of the musical symbol to be determined. These musical symbols include the stave, concert bracket for connecting staves together, tie (slur), crescendo (decrescendo), brace, beam, and tuplet. In this case, each musical symbol has its display entry point 24 not at its end point, but its beginning point so that the length data and other data for each musical symbol is stored at its beginning point.

[0017]

As for the stave, moving the cursor 25 down after specifying the beginning and the length allows a sequential display of plural staves depending on the moving distance of the cursor. The plural staves are displayed by each vertical dimension, d + e + f, of a stave area 29 to be discussed later. The lengths of these staves are unified to the same length previously specified. In this case, while moving the cursor 25 down, clicking the right button 1b of the mouse 1 on allows changing the type of the staves to be sequentially displayed. The length and the number of these plural staves may also be specified all at once by moving the cursor 25 down toward the right after specifying the beginning. The plural staves created at the same time as described above can result in simultaneous performance.

[0018]

As shown in FIG. 2, each stave line is created and displayed,

to be assigned to one row of the display parts 27. One space between two stave lines and a ledger line, and a ledger space are also created, and displayed, to be assigned respectively to one row of the display parts 27. This is intended for one-to-one correspondence between each row of the display parts 27 and a pitch of the note within the stave.

[0019]

Onto the graphic pattern 26 of the musical symbol displayed on the viewport 21, the cursor 25 appearing as the same musical symbol is moved over and clicked with the left button 1a of the mouse 1, which allows erasing the graphic pattern 26 of the displayed musical symbol. Sliding the cursor 25 appearing as a note down lower than the first stave line or up higher than the fifth stave line allows automatically displaying an additional ledger line with a space corresponding to cursor's sliding distance. This additional ledger line could be displayed unless the cursor 25 appearing as the note is moved out of the stave area 29 shown in FIG. 2. Furthermore, an entry of the bar line which marks a division of bars into a point on one of concert staves results in automatic entry of the bar line into the same point on each of the other concert staves.

[0020]

The command icons 23 are displayed as the graphic patterns 26. They indicate, from the left to right of the display screen, deletion, range specification, help, new, file, erase, enlargement, performance, print, environment, sound, and completion. The term, deletion, means deleting the displayed graphic patterns 26 of the musical symbols by separately specifying each graphic pattern to be deleted. The term, range specification, means specifying a range within the viewport 21 to erase, move or enlarge data in the specified range.

[0021]

The term, file, means saving display data and performance data of the music score composed on the viewport 21 on a floppy disk 7 as well as loading these data from the floppy disk 7. The term, erase, means erasing the displayed graphic patterns 26

of the musical symbols by specifying its erased range. The term, enlargement, means enlarging the displayed graphic patterns 26 of the music score composed on the viewport 21. The term, performance, means automatic performance based on the displayed graphic patterns 26 of the music score composed on the viewport 21.

[0022]

The term, print, means printing the graphic patterns 26 of the music score composed on the viewport 21. The term, environment, means setting the tempo and loudness for the automatic performance, the display scale and the like. The term, sound, means specifying a tone color for the music score composed on the viewport 21 or specifying the respective tone colors for performance parts of plural music scores composed for concert. These performance parts refer to melody, code and rhythm performances and those which are based on external inputs from sources such as MIDI (musical instrument digital interface).

[0023] 2. Entire Circuit

FIG. 3 shows an entire circuit in the music information display system. A mouse scan circuit 2 scans movement of the mouse 1 and clicking-on/off of the left, right and center buttons 1a, 1b, 1c. This scanning allows to detect data indicating clicking-on/off of the left, right and center buttons 1a, 1b, 1c of the mouse 1 and to write them to a RAM 13 by a CPU 11. Comparing these data to the data previously stored in the RAM 13, which also indicates clicking-on/off of the aforementioned buttons 1a, 1b, 1c, the CPU 11 determines either on or off event of the click buttons 1a, 1b, 1c occurred.

[0024]

Data indicating the horizontal (x-axis) and vertical (y-axis) displacement of the mouse 1 is added to cursor coordinate data KC in a cursor point register 137 within the RAM 13 so that the cursor 25 of the mouse 1 can move. A key scan circuit 4 scans operations of each key of a keyboard 3. This scanning allows to detect data indicating on/off operations of each key and to

write them to the RAM 13 by the CPU 11. Comparing these data to the data previously stored in the RAM 13, which also indicates on/off operations of each key, the CPU 11 determines either on or off event of each key occurred.

[0025]

FIG. 4 shows the RAM 13. The RAM 13 stores various processed data as well as the aforementioned data. The RAM 13 is provided with a display code memory 131, a stave memory 132, a performance memory 133, a scale table 134, a cursor point register 137, a search point register 138, a cursor symbol register 139, a stave number register 140, a tone color register 141, a tempo buffer 142 etc.

[0026]

The display code memory 131 stores the code data KD, including the musical symbols and letters displayed on the CRT 5, which correspond to the display point coordinate data PC. The viewport 21 is comprised of the display parts 27 divided by a unit of 3 bit length * 5 bit width. The display point coordinate data PC indicates a coordinate point by unit and row of the display parts 27. Thus, an address in the display code memory 131 consists of a row address and a column address, to be assigned to one unit of the matrix display parts 27. Unit of the display parts 27 may have any dimension equal to or larger than 1bit length * 1 bit width.

[0027]

A stave memory 132 stores data for the staves in each stave area 29 displayed on the viewport 21, which include a stave type, position and dimension, and connecting order. A set of data for each stave is stored in one of the addresses in the stave memory 132. Therefore, each address indicates the order of a certain stave, that is, the performance order for the certain stave. The stave type to be indicated includes a five-line type and a single-line type. The stave position indicates the display point coordinate data PC of the display entry point 24 of the stave. For example, the first stave shown in FIG. 1 is indicated as (x, y) = (g + a, h + c).

[0028]

The stave dimension indicates a dimension of the stave area 29. For example, the dimension of the first stave shown in FIG. 1 is indicated as (a + b + c), (d + e + f). In case of the single-line type of the stave, the aforementioned (e) is set to "0". The connecting order of the staves indicates the order from the beginning of the staves connected together. A group of the staves connected together having their connecting order data stored can result in simultaneous performance. If no staves connected together are simultaneously created, the connecting order data is set to "0".

[0029]

The performance memory 133 stores music performance data MP of a converted data format from the code data KD for the respective musical symbols within the stave area 29 arranged in order of performance. The scale table 134 stores note number data NT determined depending on the y-axis coordinate of the display point coordinate data PC by unit of the display parts 27 in the stave area 29. Each note number data NT varies depending on the clef.

[0030]

The cursor point register 137 stores the cursor coordinate data KC, which indicates a pointer coordinate (x, y) of the cursor 25 on the display screen 20. The search point register 138 stores the search coordinate data SC, which indicates a pointer coordinate (x, y) of data search in the display code memory 131 or the stave memory 132. The cursor symbol register 139 stores the code data KD for the musical symbols displayed as the cursor 25. The cursor 25 appearing as a shape of the musical symbol based on the code data KD is displayed at a pointer (x, y) of the cursor coordinate data KC. If no code data KD is established, the arrow cursor 25 appears.

[0031]

The stave number register 140 stores the stave number data SN for indicating the order of plural staves displayed in the viewport 21. The tone color register 141 stores the tone color

data for each performance part, which was preset with the sound command icon 23. The tempo buffer 142 stores the tempo data, which was preset with the environment command icon 23 or which is displayed on the music score within the viewport 21.

[0032]

The RAM 13 also stores the graphic data GF based on the code data KD for the musical symbols and character data based on the letter code. The code data KD for the musical symbols is converted to this graphic data or the like and written to the graphic memory 6. This allows displaying the graphic patterns 26 of the musical symbols, letters, etc. on the CRT 5. The RAM 13 further stores programs for the CPU 11 to execute various processes shown in a flowchart to be described later.

[0033]

The graphic data for the symbol icons 22 and the command icons 23 is written to the graphic memory 6 to be displayed on the CRT 5. The programs and data, which are stored in the RAM 13, were originally stored in the floppy disk 7 and then read out via the floppy disk driver 8 to be transmitted to the RAM 13, while these programs and data are in turn read from the RAM 13 to be stored in the floppy disk 7. The floppy disk 7 may be replaced with another storing means such as an optical memory and a memory card. The ROM 12 may store programs and the like for the CPU 11 to execute various processes shown in a flowchart to be described later and other processes as required. The contents stored in the graphic memory 6 are read out by the CPU 11 and printed by the printer 9 via a print driver 10.

[0034]

The tone generator 14 in FIG. 3 generates tone signals based on the pitch, tone color, tone length, velocity and the like according to the music performance data MP in the performance memory 133. These signals are sent to the sound system 15 to generate tones. The tone generator 14, in which a plural number of, for example, 16 tone generating channels are formed by the time-dividing process, can generate polyphonic tones. An unillustrated assignment memory stores data concerning the

tones to be assigned to these channels. The tone generator 14 and the sound system 15 may be provided with an interposed MIDI interface (not shown). The tone generation may also be accomplished based on the performance data to be separately inputted via the MIDI interface.

[0035] 3. Data Format

FIG. 5 shows the code data KD for the musical symbols stored in the display code memory 131. FIG. 6 shows the music performance data MP converted from this code data KD for the musical symbols and stored in the performance memory 133. The code data KD for each musical symbol is comprised of content data and additional code for the musical symbol. The content data shows what the musical symbols are. The examples are as follows: a blank; clef including G clef, f clef and C clef; note including upward quarter note, downward quarter note and quarter rest; stave type; accidental mark including sharp, flat and natural; denominator number of the time signature; dynamic mark including fortissimo ff, forte f and pianissimo pp; respective concert brackets; respective bar lines; respective repeat marks; tie; crescendo and decrescendo; brace; respective articulations; respective performance marks; and number of the tempo mark.

[0036]

The additional code is provided by setting an additional flag bit if the content data is not sufficient to describe what a musical symbol is. For example, the additional code for note tells whether or not a note overlaps other musical symbols, the presence or absence of a ledger line, whether or not the notes constitute a chord, and the presence or absence of a beam and a tuplet. In addition to that, the additional code for time signature tells the presence or absence of a numerator of the time signature, the presence or absence of length data for the stave, tie, crescendo, decrescendo and the like, the order to create staves, the presence or absence of staves connected together, and whether or not a stave is at the beginning of the staves connected together. Based on this additional code, the

display code memory 131 stores additional data to be discussed later.

[0037]

The additional data stored contains the length data for the numerator number of the time signature, stave, tie, crescendo, decrescendo and the like. Furthermore, data for the beam, which contains a type and position of the beam, is stored as the additional data. The beam type data indicates the number and the shape of the beams as shown by A, B, C and D in FIG. 9. The position data on the beam indicates where the applicable note is positioned either at the beginning, middle or end of the beam. These beginning, middle and end of the beam correspond to A, B and C, and D in FIG. 9, respectively.

[0038]

The additional data for the stave contains the order to create staves, and the width in addition to the aforementioned length data. The data for the order to create staves indicates the order by which each stave is created. In the case of the staves created at the same time and connected together, these staves have common data to be stored as the order to create staves. This order data for the beginning one of the staves created at the same time and connected together is used as the common data. A group of the staves connected together having this common data stored as the order data to create staves can result in simultaneous performance. The width data indicates the entire width of the staves connected together.

[0039]

Furthermore, the presence or absence of a ledger line for the stave, to which the applicable note belongs, is stored as additional data. Sliding the cursor 25 appearing as a note down lower than the first line of the stave results in data, indicating that the applicable note belongs to the stave located above this note, being stored as additional data. In turn, sliding the cursor 25 appearing as a note up higher than the fifth line of the stave results in data, indicating that the applicant note belongs to the stave located below this note,

being stores as additional data.

[0040]

The code data KD for the display musical symbols, which is converted into the data for performance, is referred to as the music performance data MP. The music performance data is converted as shown in FIGs. 13 and 14. The music performance data MP includes step time data ST, gate time data GT, sound-on/sound-off command data KM, note number data NT, and velocity data BR.

[0041]

The step time data ST indicates time length from the beginning of a tune or a bar to sound-on. The gate time data GT indicates time length from sound-on to sound-off. The command data KM indicates the sound-on/sound-off commands. The note number data NT indicates pitch. The velocity data BR indicates how fast or how hard a key is operated to generate or dampen a tone. Converting the code data KD for the musical symbols into the music performance data MP may involve creating intermediate data as required. The music performance data MP is converted via this intermediate data.

[0042] 4. Entire Processes

FIGs. 7 and 8 show a flowchart for the entire processes. These processes are started at turn-on, or at the mode selection by key operations after power is turned on. The initializing process is implemented by the CPU 11 to clear the display code memory 131, the stave memory 132, the performance memory 133, the scale table 134, the cursor point register 137, the search point register 138, and the cursor symbol register 139 in the RAM 13 (Step 01). Next, the graphic data GF for the symbol icons 22, the command icons 23, the viewport 21 and the like shown in FIG. 1 is written to the graphic memory 6 to be displayed on the CRT 5 based on the program stored in the ROM 12 (Step 02).

[0043]

If it is found that the mouse 1 is moved based on the data from the mouse scan circuit 2, data corresponding to the

displacement of the mouse 1 is added to or subtracted from the cursor coordinate data in the cursor point register 137 within the RAM 13 (Steps 03 and 04). Then, if it is found that the left button 1a of the mouse 1 is clicked on based on the data from the mouse scan circuit 2 (Step 05), it is determined whether or not the cursor coordinate data is on the viewport 21 of the CRT 5 (Step 06), then whether or not it is on a symbol icon 22 (Step 07), then whether or not it is on a "performance", "sound" or "file" command icon 23 (Steps 08, 09 and 10), and then whether or not it is on another command icon 23 (Step 11).

[0044]

If it is determined that the cursor coordinate data is on a symbol icon 22 (Step 07), the graphic data GF for the arrow cursor 25 written to the address in the graphic memory 6 is erased. Then, the code data KD for the musical symbol applicable to this symbol icon 22 is written to the cursor symbol register 139 in the RAM 13, and the graphic data GF corresponding to this musical symbol for the symbol icon 22 is written to the graphic memory 6 (Steps 12 and 13). This allows the arrow cursor 25 for indicating a position, to which the mouse 1 is moved, to turn to a musical symbol. The cursor 25 appearing as the musical symbol is moved on the display screen 20 of the CRT 5, depending on the displacement of the mouse 1 in the Step 03 and Step 04.

[0045]

If it is determined that the left button 1a of the mouse is clicked on the viewport 21 (Step 06), and furthermore that, the cursor 25 of the mouse 1 turns to a musical symbol, in other words, the code data KD for this musical symbol is stored in the cursor symbol register 139 (Step 14), this allows executing the process to write this code data KD for the musical symbol stored in the cursor symbol register 139 to an address based on the cursor coordinate data KC in the display code memory 131 (Step 15), to write the graphic data GF for this musical symbol to the graphic memory 6 to be displayed on the CRT 5 (Step 16).

[0046]

If it is determined that the left button 1a of the mouse is

clicked on a "performance" command icon 23 (Step 08), search/performance is processed as will be described later (Step 17). If it is determined that the left button 1a of the mouse is clicked on a "sound" command icon 23 (Step 09), tone color setting for the performance parts is processed (Step 18). In this case, the music score for concert consists of the performance parts to be displayed on the window as "part 1 = ," "part 2 = ," ... "rhythm 1 = ," ... and "MIDI 1 = ," ... to which the tone color number data is respectively entered by inputting a number from the keyboard 3. The tone color data is therefore set into the tone color register 141 in the RAM 13.

[0047]

If it is determined that the left button 1a of the mouse is clicked on a "file" command icon 23 (Step 10), record of the data to the floppy disk 7 or the reproduction from the floppy disk 7 is processed (Step 19). This data to be recorded includes the music performance data MP in the performance memory 133 and the code data KD for the musical symbols in the display code memory 131 within the RAM 13. If it is determined that the left button 1a of the mouse is clicked on a command icon 23 other than the above (Step 11), a certain process is performed based on this command icon (Step 20).

[0048]

If it is determined that the right button 1b of the mouse 1 is clicked on (Step 21), and further that, the cursor 25 of the mouse 1 turns to a musical symbol, in other words, the code data KD for this musical symbol is stored in the cursor symbol register 139 (Step 22), this allows determining whether or not this musical symbol indicates a beam (Step 23). If it is determined that the musical symbol indicates a beam, a flag direction of this beam can be changed by changing the additional data for indicating this flag direction contained in the code data KD for this beam (Step 24).

[0049]

Therefore, as shown in FIG. 9, when the flags of four notes, e.g., an eighth note, a sixteenth note, an eighth note, and a

sixteenth note, are connected to became a beam, clicking the right button 1b of the mouse 1 on can change the beam direction of the second note, a sixteenth note. In this case, the code data KD itself for the beam may be changed by clicking the right button 1b of the mouse 1 on.

[0050]

Also in the Steps 21 through 23, the code data KD for the musical symbols except beam is processed to allow the shape of the cursor 25 appearing as a musical symbol to change in a cyclic manner each time clicking the right button 1b of the mouse 1 on (Step 25). In the case of the code data KD for a musical symbol, e.g., a note (rest), in the cursor symbol register 139, this note sequentially changes to upward quarter note, then downward quarter note, quarter rest, and again upward quarter note, then… in a cyclic manner. This cyclic change is based on a count of the number in the content data contained in the code data KD.

[0051]

If all the code data KD for the quarter note, downward quarter note and quarter rest have the same bit configuration except a specific bit, such as a least significant bit, a value of "1" could be added to or subtracted from the specific bit. In order to enter these notes, pressing a numeric keypad "1" of the keyboard 3 allows the note to become a single-dotted note and pressing a numeric keypad "2" allows the note to become a double-dotted note, thereby changing the code data KD for the musical symbols.

[0052]

If it is determined that the center button 1c of the mouse 1 is clicked on (Step 26), and further that, this clicking is on the viewport 21 (Step 27), the graphic data GF based on all the coordinate points, (x, y) in which x remains unchanged while y is variable, for the cursor coordinate data KC in the cursor point register 137, is written to the graphic memory 6 (Step 28).

[0053]

This allows the cursor 25 to draw an auxiliary line 28 as a musical symbol on the display screen 20 of the CRT 5 as shown in FIG. 2. This can help draw the musical symbols in vertical alignment on the music score. Colors between the auxiliary line 28 and the musical symbols may not be necessarily the same. The auxiliary line may also be formed into any shape, such as horizontally long, L-shape, T-shape, cross and box. In such case, writing of the graphic data GF to the graphic memory 6 is processed based on all the coordinate points (x, y) in which x is variable while y remains unchanged.

[0054]

Displaying the auxiliary line 28 may also be achieved by any operations including pressing any key on the keyboard 3 other than clicking the center button 1c of the mouse 1 on in the Step 21. The auxiliary line 28 could be displayed not only for the aforementioned graphic patterns 26 of the musical symbols, but also displayed for other graphic patterns.

[0055]

In the Step 12, sliding the cursor 25 appearing as a note down lower than the first line of the stave or up higher than the fifth line allows additional data for a ledger line to be written to the code data KD for the applicable note. This results in additional graphic pattern 26 of the ledger line to the graphic pattern 26 of the note located lower than the first stave line or the higher than the fifth line.

[0056]

In such case, the addition of the ledger line is limited to a range within the stave areas 29. If plural stave areas 29 overlay one another, the additional ledger line is provided for one of these staves, to which the cursor 25 appearing as a note is moved. This is because the data for the stave to which the applicable note belongs is also stored in the additional data. Based on this data, the direction in which the additional ledger line is displayed is determined. Some number of the ledger lines are each displayed every second row of the display parts 27 corresponding to a space between the applicable note and the

first or fifth stave line. The number of the ledger lines could be counted depending on cursor's sliding distance from the stave to which the applicable note belongs. This may be stored in the additional data.

[0057] 5. Musical symbol writing process

FIG. 10 shows a flowchart of the musical symbol writing process in the Step 15. This writing process determines whether or not the cursor 25, which was determined in the Step 14 that it turned to a musical symbol, appears as either one of the musical symbols that require to specify the beginning and end, such as a stave, concert bracket for connecting staves together, tie (slur), crescendo (decrescendo), brace, beam, and tuplet (Step 101).

[0058]

If determined Yes, and the beginning is not yet specified (Step 102), the cursor coordinate data KC for the cursor 25 when the left button 1a of the mouse is clicked on in the Step 05 is temporally stored in the RAM 13, thereby specifying the beginning (Step 104). In contrast, if the beginning is already specified in the Step 102, and the code data KD in the cursor symbol register 139 does not refer to the staves (Step 103), a type, display point coordinate data PC and length of the musical symbol are temporally stored in the RAM 13, thereby specifying the end (Step 105). The cursor coordinate data KC itself is used as this display point coordinate data PC.

[0059]

In the Step 103, the code data KD in the cursor symbol register 139 refers to the staves, and the beginning and end of the staves are already specified (Step 103), data for a type, length and order of the staves is temporally stored in the RAM 13, thereby specifying the bottom edge of these plural staves (Step 106). This allows creating the plural staves at the same time. Whether or not the beginning, end or bottom edge is already specified is stored in the RAM 13 as flag data.

[0060]

If the cursor 25, which was determined in the Step 14 that it turned to a musical symbol, appears as a beam (Step 107),

and furthermore the beginning and end are already specified (Step 108), writing/erase of the code data KD for the musical symbol are processed in the next Steps 109 through 111. The beam direction can be changed in the Step 24. Changing it to a desired direction, clicking the left button 1a of the mouse on allows this direction to be set.

[0061]

Next, in an attempt to write the new code data KD for the musical symbol to the address in the display code memory 131 based on the display point coordinate data PC, if the code data KD for the same musical symbol is already written into this address (Step 109), the new code data KD is erased (Step 110).

[0062]

If the code data KD for the same musical symbol is not yet written (Step 109), each code data KD temporally stored in the RAM 13 or the code data KD stored in the cursor symbol register 139 is written (Step 111). In such a case that the new code data KD for the musical symbol to be written is different from the code data KD already written, this new code data KD is written into a preliminary area in the display code memory 31 or written as preliminary data.

[0063]

Also, the code data KD for the bar line, which marks a division of bars on one of the concert staves, is written to a point in the display code memory 131, which is followed by automatic writing of this code data KD for the bar line on the other concert staves respectively to the same points as before. Relative to the address based on the coordinate (x, y) to which the code data for the bar line on one of the concert staves is written, the addresses, to which this code data for the bar line on the other concert staves is written, are based on coordinates (x, y + d + e + f), (x, y + 2d + 2e + 2f) and (x, y + 3d + 3e + 3f) etc. In such case, if plural stave areas 29 overlay one another, the y coordinate data for the display entry points 24 of these staves is used. However, writing of this code data is not allowed for the other concert staves while prescribed

operations are going on, such as pressing on any key on the keyboard 3.

[0064]

As described above, the musical symbols can be entered into and displayed at their respective entry points. Also, any length can be specified for the musical symbols, which include the stave, concert bracket for connecting staves together, tie (slur), crescendo (decrescendo), brace, beam, and tuplet. Furthermore, the clef can be entered and displayed separately from the stave, to compose any music score.

[0065] 6. Search/performance processes

FIG. 11 shows a flowchart of the stave search process, which is included in the flowchart of the search/performance processes in the Step 17. This process allows the CPU 11 to clear the search coordinate data SC in the search point register 138 in the RAM 13 in order to set the search coordinate data SC, (x, y) = (0, 0), while clearing the stave number data SN in the stave number register 140 (Step 31), thereby reading out the data at the address based on the coordinate (x, y) = (0, 0) in the display code memory 131 (Step 32).

[0066]

If the data read out is not the code data KD for the stave (Step 33), the x coordinate for the search coordinate data SC is set to x + 1 (Step 34), and the search process of the code data KD for the stave is carried out in the Steps 32 and 33 until the x coordinate for the search coordinate data SC reaches the right end of the display screen 20 (Step 35). If the search by one row (y = 0) is completed (Step 35), the x coordinate for the search coordinate data SC is set to "0", as well as the y coordinate for the search coordinate data SC to y + 1 (Step 36), and the same search processes (Steps 32 and 33) continue until the (x, y) coordinate for the search coordinate data SC reaches the end of the display screen 20 (Step 37).

[0067]

If this search results in reading out of the code data KD for the stave (Step 33), a bit in the additional code contained in

the code data KD for this stave is identified which it indicates the presence or absence of other staves connected together, to determine whether or not the staves simultaneously created for simultaneous performance are connected together (Step 38). If it is determined that the stave is not connected with other staves, the stave type and length data stored at the searched address in the display code memory 131 is read out to create the dimension and position data, which is written to the address based on the stave number data SN in the stave memory 132 (Step 39), thereby setting the stave number data SN to SN + 1 (Step 40).

[0068]

The stave type includes a five-line type and a single-line type. The stave position is represented as the search coordinate data SC, (x, y), that is, the display point coordinate data PC for the display entry point 24 of the stave. The stave dimension is represented by a dimension of the stave area 29 as (a + b + c) * (d + e + f). The stave number data SN, which is stored in the stave number register 140, indicates the addresses to which the various data for the respective staves are written, and the orders to perform these staves. The aforementioned (a), (c), (d), (e) and (f) are predetermined depending on the stave type, while (b) uses the stave length data itself. Any number equal to or greater than "0" is selected as the values of (a), (c), (d), (e) and (f). In such case, the stave areas 29 may overlay one another, depending on a value setting of (d) and (f).

[0069]

If it is determined in the Step 38 that the searched stave is connected with other staves, a bit in the additional code contained in the code data KD for this stave is identified whether or not it indicates the beginning of the staves connected together (Step 41). If it is determined that the bit indicates the beginning of the staves connected together, the stave type and length data stored at the searched address in the display code memory 131 is read out to create the dimension

and position data, which is written to the address based on the stave number data SN in the stave memory 132 (Step 42). This is followed by writing the connecting order data, "1," (Step 43) to set the stave number data SN to SN + 1 (Step 44).

[0070]

The data at an address in the display code memory 131 is read out. This address is based on a coordinate obtained by adding the stave dimension data, "d + e + f," to the y coordinate of the search coordinate data SC, (x, y). If it is determined that there are other staves connected together (Step 45), the type, position and dimension data for one of these staves, stored at the searched address, is read out to be written to the next address in the stave memory 132 in the same manner as the Steps 42 through 44. This is followed by writing the connecting order data, "2" (Steps 42 through 44).

[0071]

Then, the data at addresses in the display code memory 131 (Step 45) is read out. These addresses are based on coordinates respectively obtained depending on each stave dimension data "d + e + f" to be read out. The type, position and dimension data of the other staves are written to the stave memory 132 in the same manner as previously noted. This is followed by writing the connecting order data, "3", "4", "5" etc. (Steps 42 through 44).

[0072]

This process continues to the width data for the staves connected together (Step 45). When the process is completed, the search coordinate data SC is returned to (x, y), that is, the display coordinate data PC for the display entry point 24 of the beginning one of the staves connected together (Step 46). This prevents overwriting the data to the stave memory 132 (Step 41) even if the staves connected together are re-searched in the next search.

[0073]

All of the staves are searched no matter if they are connected or not, followed by searching the connected portions of the

staves connected together and setting a flag indicating connection for the respective staves. The staves provided with this flag may involve adjustment of the step time data ST in the Step 77 to be discussed later.

[0074]

These search processes are performed for all the code data KD for the staves stored in the display code memory 131. If the searches at all the addresses in the display code memory 131 completed (Step 37), the musical symbol search process is performed for the code data KD for the musical symbols transmitted to the stave memory 132 (Step 47).

[0075]

These searches are performed in the x-axis direction repeatedly. The upper one of the plural staves that overlay on the display screen has priority to be searched. Also the left one of the plural staves displayed on the same row has priority to be searched. Automatic performance is sequentially executed based on this search order. Start and end points of the searches are at not only (0, 0) and (END, END) on the viewport 21, respectively, but these points may also be at (0, END) and (END, 0) or at (END, END) and (0, 0). The searches may be performed in the y-axis direction as with the case of after-mentioned musical symbol search process, or in the reverse direction of the usual x-axis or y-axis direction.

[0076]

The aforementioned search/performance processes may be performed on the viewport 21 in a range of the music score specified by using the command icon 23 for range specification. In such case, the search coordinate data SC (x, y) is not (0, 0) set in the Step 31, but a coordinate data at the left top corner in the specified range. The coordinate data to be identified in the Steps 35 and 37 is an x coordinate data at the right end of the specified range, and a coordinate data at the right bottom corner in the specified range, respectively. The x coordinate data to be set in the Step 36 is an x coordinate data at the left end of the specified range.

[0077] 7. Musical symbol search process

FIG. 12 shows a flowchart of the musical symbol search process in the Step 47. This process allows the CPU 11 to read out the first stave position data for the stave number data SN = 0 in the stave memory 133, that is, the display point coordinate data PC for the display entry point 24 to be set as the data search coordinate data SC (x, y) (Step 51). This is followed by reading out the code data KD for the musical symbol at an address based on the coordinate (x, y) in the display code memory 131 (Step 52).

[0078]

If the data read out is not the code data KD for the musical symbol (Step 53), the y coordinate for the search coordinate data SC is set to y + 1 (Step 54), and the search process for the musical symbol is carried out in the Steps 52 and 53 until the y coordinate for the search coordinate data SC reaches "d + e + f," that is, the bottom end of the stave area 29 (Step 55). If the search by one column (x = 0) is completed (Step 55), the y coordinate for the search coordinate data SC is set "0", as well as the x coordinate for the search coordinate data SC to x + 1 (Step 56), and the same search processes (Steps 52 and 53) continue until the (x, y) coordinate is equal to (a + b + c, d + e + f), that is, this coordinate reaches the end of the stave area 29 (Step 57).

[0079]

If this search results in reading out of the code data KD for the musical symbol (Step 53), the process to convert this code data into the music performance data MP is performed (Step 58). If the search is completed to the end of the stave area 29 in the Step 57, it is determined whether or not there is any code data KD for other staves (Step 59). If determined YES, the search processes in the Steps 51 through 58 are repeated. If determined NO, the automatic performance process is performed based on the music performance data MP written to the performance memory 133 (Step 60).

[0080]

In the automatic performance process, the music performance data MP, to be described later, is read out from the performance memory 133, and sound-on/sound off is executed by a lapse of the step time data ST and by a lapse of the gate time data GT, respectively, at a speed based on the tempo data in the tempo buffer within the RAM 13. In such case, the musical tone color depends on the tone color data for each performance part stored in the tone color register within the RAM 13, while the pitch depends on the note number data NT contained in the music performance data MP, and the velocity such as loudness depends on the velocity data BR contained in the music performance data MP. In this case, the tone color data, which is determined by the aforementioned "sound" command icon 23 and stored in the tone color register within the RAM 13, may be written to the performance memory 133 as one of the music performance data MP in the after-mentioned Steps 79 and 80.

[0081] <u>8. Performance data creation process</u>

FIGs. 13 and 14 show a flowchart of the music performance data creation process in the Step 58. This process allows the CPU 11 to create the music performance data MP based on the code data KD for the musical symbols searched in the Step 53. If the searched musical symbol indicates a clef (Step 61), the note number data NT is created, which corresponds to each y-axis coordinate of the display point coordinate data PC for the display parts 27 in the stave area 29 to be written to the scale table 134 within the RAM 13 (Step 62). As for the display point coordinate data PC, if this clef indicates G clef, the display point coordinate data PC for the first stave line corresponds to the note number data NT for the pitch E4, while the display point coordinate data PC for the first stave space corresponds to the note number data NT for the pitch F4, and the display point coordinate data PC for the second stave line corresponds to the note number data NT for the pitch G4.

[0082]

If the searched musical symbol indicates a key signature or an accidental mark (Step 63), the note number data NT in the

scale table 134 based on the display point coordinate data PC for this musical symbol is raised a half step up or lowered a half step down (Step 64). If the searched musical symbol indicates a bar line which marks a division of bars (Step 65), the scale table 134, which was changed in the Step 64, is restored (Step 66). However, if the musical symbol indicates a key signature, the scale table changed is not restored. Each total time data of the step time data ST and the gate time data GT towards the bar line in the music scores for concert is adjusted to the longest total time data by increasing the value of the next step time data ST in each music score (Step 67).

[0083]

If the searched musical symbol indicates a tempo mark (Step 68), the tempo data TP for this tempo mark is set into the tempo buffer 142 within the RAM 13 (Step 69). This allows the speed of the automatic performance in the Step 50 to be determined. This tempo data TP may also be written to the beginning area of the music performance data MP in the performance memory 133.

[0084]

If the searched musical symbol indicates a dynamic mark (Step 70), the velocity data BR is created based on a value of this dynamic mark (Step 71). There are plural dynamic marks indicating various levels of loudness from pianissimo pp to fortissimo ff. In this case, if the code data KD for the musical symbol, crescendo/decrescendo, is searched (Step 72), the velocity data BR is created based on the values between the beginning and end of the dynamic mark, crescendo/decrescendo, which are calculated in proportion to the number of notes (Step 73). If no dynamic mark is indicated, the preset data is used as the velocity data BR (Step 74). The aforementioned proportional conversion may be replaced with any other calculations based on the arithmetic expressions stored in the RAM 13 or the like.

[0085]

If the searched musical symbol indicates a note (Step 75), the display point coordinate data PC for this musical symbol

is converted in the scale table 134 which was created or changed in the Steps 62, 64 and 66 into the note number data NT (Step 76). Next, setting the step time data ST at a predetermined constant value (Step 77), the gate time data GT, that is, the note length according to the type of this note, is calculated based on its articulation such as staccato and tenuto (Step 78).

[0086]

The step time data ST, the note number data NT and the velocity data BR are written into the performance memory 133 in addition to the sound-on command (Step 79), followed by writing the gate time data GT and the note number data NT into the performance memory 133 in addition to the sound-off command (Step 80). The velocity data BR for the sound-off command is set at "0". However it may be set at any value.

[0087]

If the connecting order data is already written for the staves connected together in the Steps 43 and 44, the step time data ST to be determined in the Step 77 is unified. If no connecting order data is written, the step time data ST is so determined as to start sound generation immediately after the sound generation of the last note on the previous stave. Tunes on the respective staves within the viewport 21 may be sequentially performed in the tune order stored in the music performance data MP in the Step 60. In such case, the same tune order is set for the staves connected together.

[0088]

If the code data KD for the searched musical symbol indicates a rest (Step 81), the time data, that is, the rest length according to the type of this rest, is added to the next step time data ST in the music performance data MP (Step 82).

[0089]

If the code data KD for the searched musical symbol indicates a tie (Step 83), each time data, that is, each note length of the respective notes to the end of this tie is accumulated to the gate time data GT for the sound-on command at the beginning of the tie (Step 84).

[0090]

If the code data KD for the searched musical symbol indicates a tuplet (Step 85), the sound-on/sound-off commands by the number of notes in tuplet are written to the performance memory 133, and each of the gate time data GT for the whole length of the tuplet, divided by the number of tuplet notes, is written to the performance memory 133, to respectively correspond to the aforementioned sound-on/sound-off commands, while the step time data ST and the velocity data BR are set at predetermined constant values (Step 86).

[0091]

If the code data KD for the searched musical symbol indicates another musical symbol (Step 87), the data for this musical symbol is written to the performance memory 133 (Step 88). If these conversion processes complete for all the code data KD for the musical symbols stored in the stave memory 132 (Step 89), the processes return to the top. The code data KD for the musical symbols can therefore be converted into the music performance data MP, providing the automatic performance based on this music performance MP in the Step 60.

[0092]

The present invention is not limited to the above embodiment, and various changes and modifications can be made without departing from the scope of the invention. For example, if the code data KD for the stave symbol is found in the Step 33, it may be possible to jump to the Steps 51 through 57 to execute the musical symbol search process and then go back to the Step 34. In addition, the dimension data for the staves connected together may be processed as the entire dimension of all the stave areas 29 connected together. The search for the musical symbols is not designed to search the code data KD for the musical symbols, but to search the graphic data GF for the musical symbols. If the graphic data GF for the musical symbols is found, it may be compared to each graphic data in the RAM 13 in order to identify the corresponding code data KD for the musical symbols. Furthermore, in place of clicking the buttons 1a, 1b,

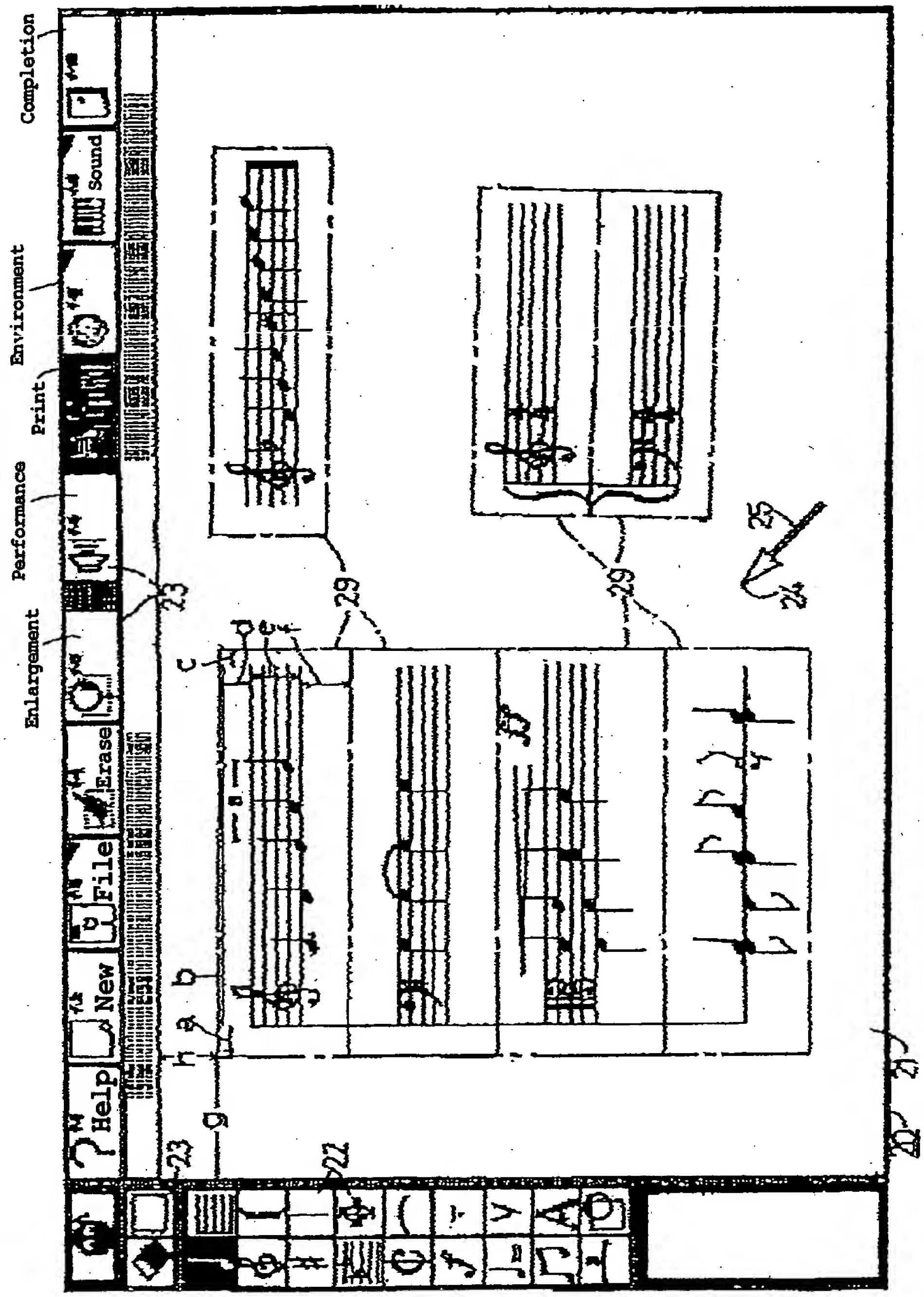
1c of the mouse 1 in the Steps 05, 21 and 26, any operations may be allowed, such as pressing on the cursor keys and enter key on the keyboard 3. One to one correspondence between each display part 27 and each displayed stave line or space may not be necessary. Instead, a single line or space may be assigned to the plural display parts 27.

[0093]

[Effect of the Invention]

As described above, the music information display system of the present invention sequentially searches the displayed musical symbol information and arranges them in order from the beginning of the musical score to create the music performance information. This allows displaying the music score while allowing one to listen to the performance based on this displayed music score, resulting in significant effectiveness in music composition and education. Entries of various musical symbol information into each display part allows a display of the graphic patterns corresponding to each musical symbol information across the plural display parts. Therefore, each musical symbol can be entered into and displayed at any entry point. The lengths of the musical symbols such as stave can be specified to any value. Also the clef may be entered and displayed separately from the stave. This allows composition of any music score.

FIG. 1



Help
New
File
Erase
Enlargement
Performance
Print
Environment
Sound
Completion
20
21
22
23
24
25
29
a
b
c
d
e
f
g
h

FIG. 2

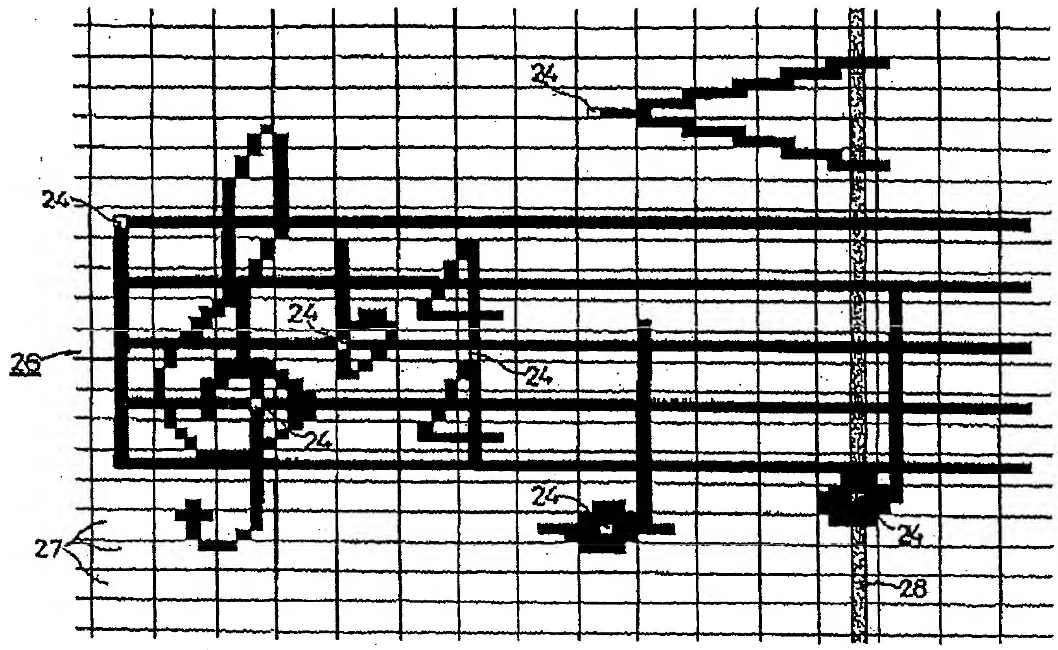

24
24
24
24
24
24
26
24
24
27
28

FIG. 3

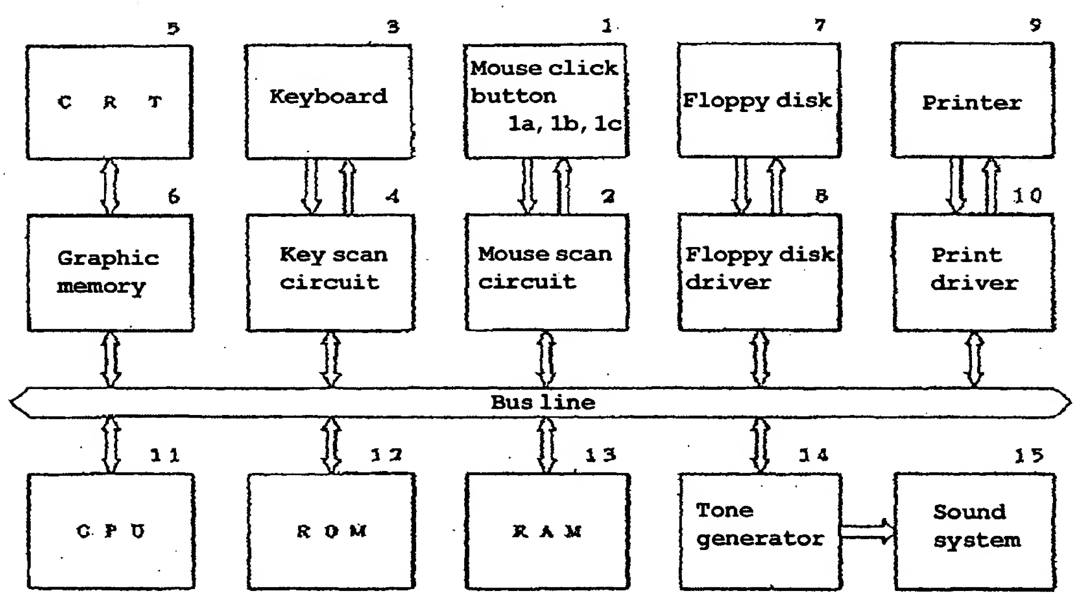

5
C R T
3
Keyboard
1
Mouse click button 1a, 1b, 1c
7
Floppy disk
9
Printer
6
Graphic memory
4
Key scan circuit
2
Mouse scan circuit
8
Floppy disk driver
10
Print driver
Bus line
11
CPU
12
ROM
13
RAM
14
Tone generator
15
Sound system

FIG. 4

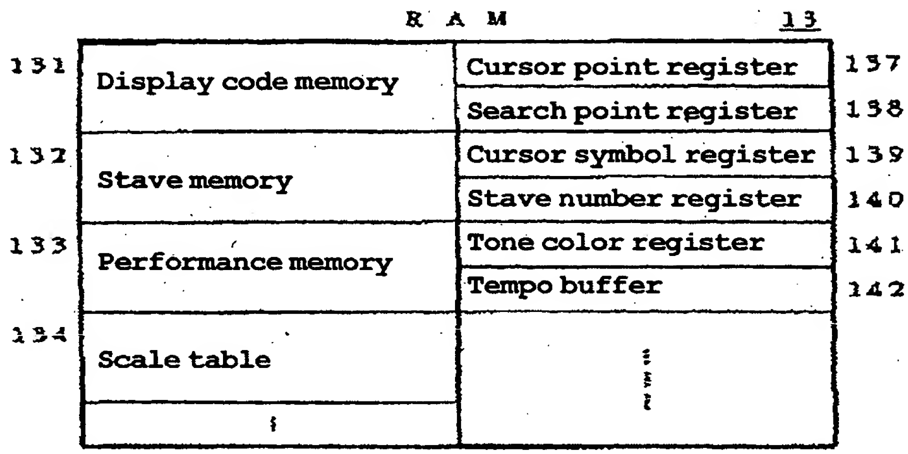


FIG. 5

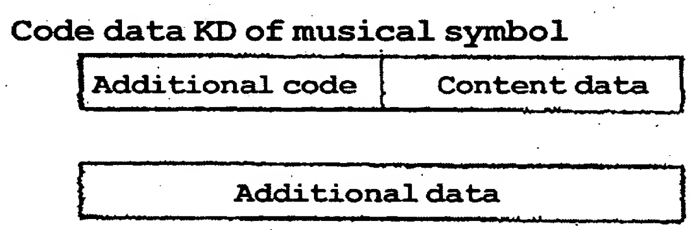


FIG. 6

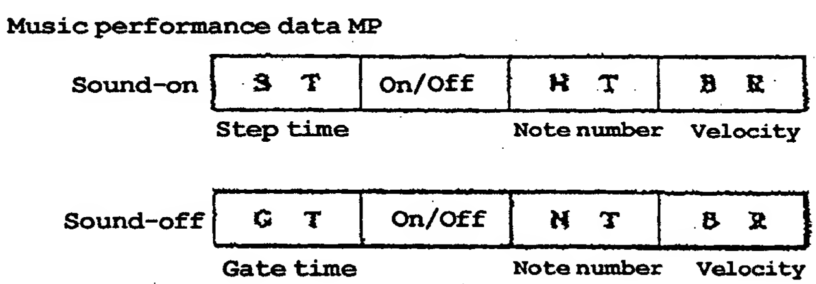
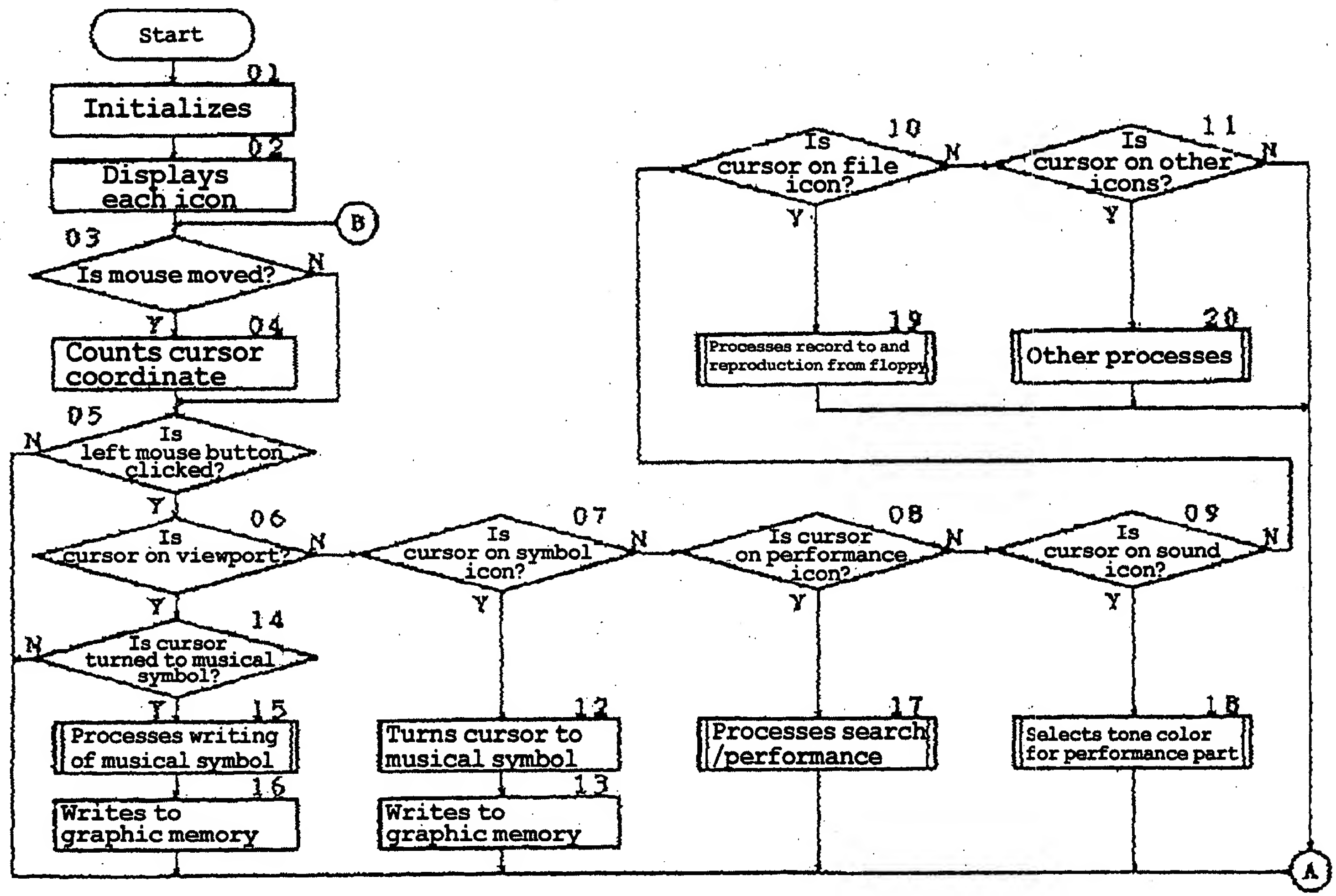


FIG. 7



Start
01
Initializes
02
Displays each icon
B
03
Is mouse moved?
Y
N
04
Counts cursor coordinate
05
Is left mouse button clicked?
Y
N
06
Is cursor on viewport?
Y
N
07
Is cursor on symbol icon?
Y
N
08
Is cursor on performance icon?
Y
N
09
Is cursor on sound icon?
Y
N
10
Is cursor on file icon?
Y
N
11
Is cursor on other icons?
Y
N
12
Turns cursor to musical symbol
13
Writes to graphic memory
14
Is cursor turned to musical symbol?
Y
N
15
Processes writing of musical symbol
16
Writes to graphic memory
17
Processes search /performance
18
Selects tone color for performance part
19
Processes record to and reproduction from floppy
20
Other processes
A

FIG. 8

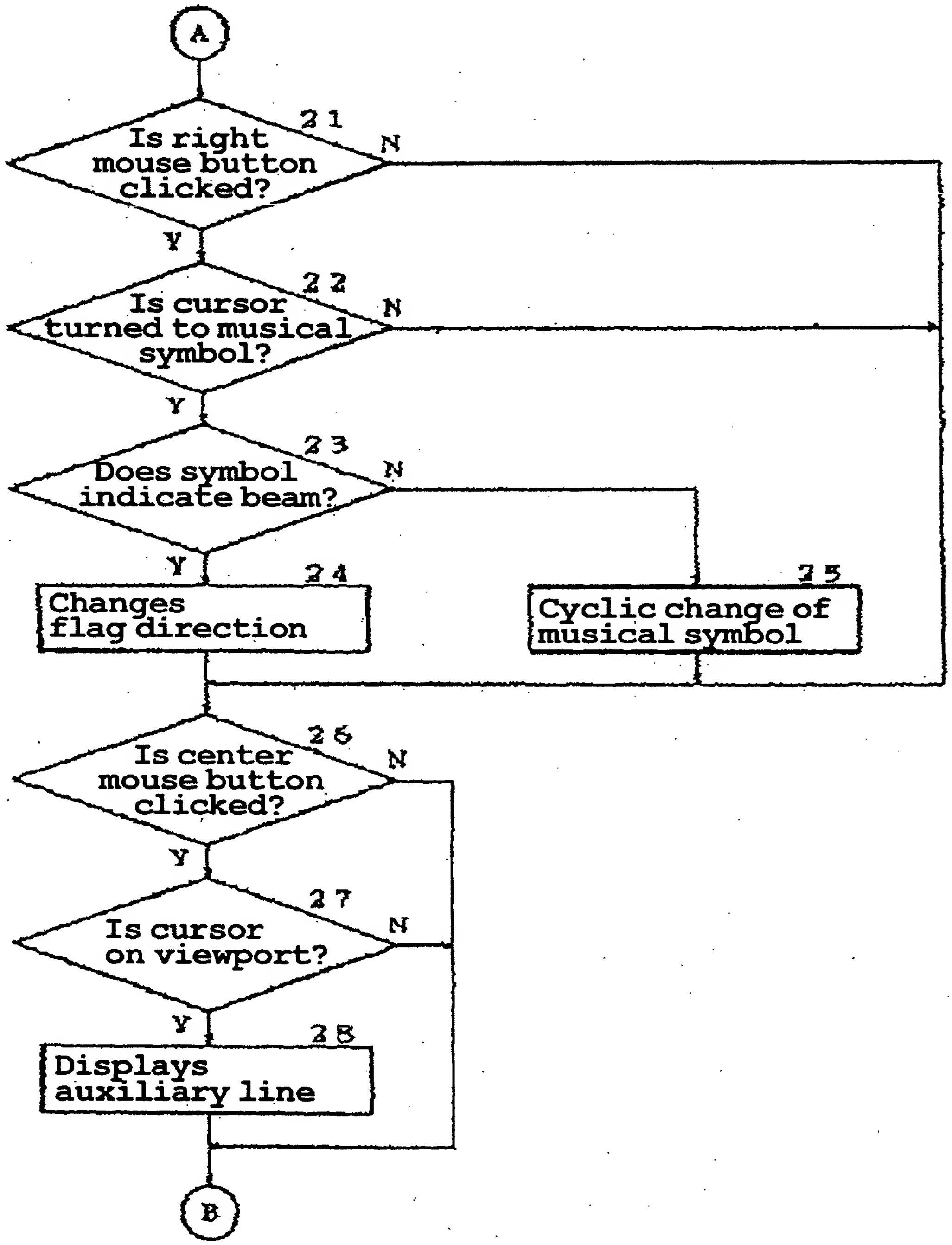
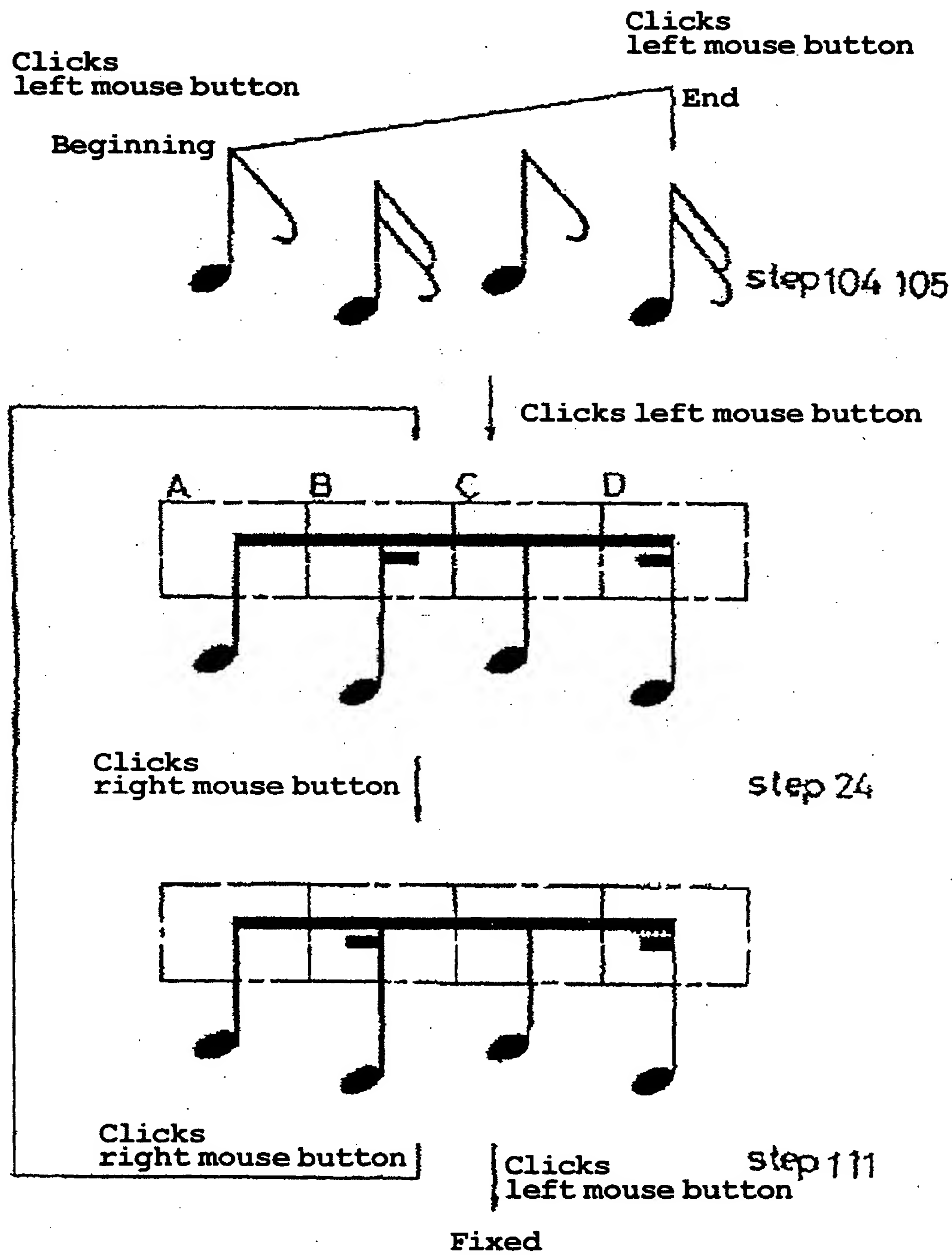
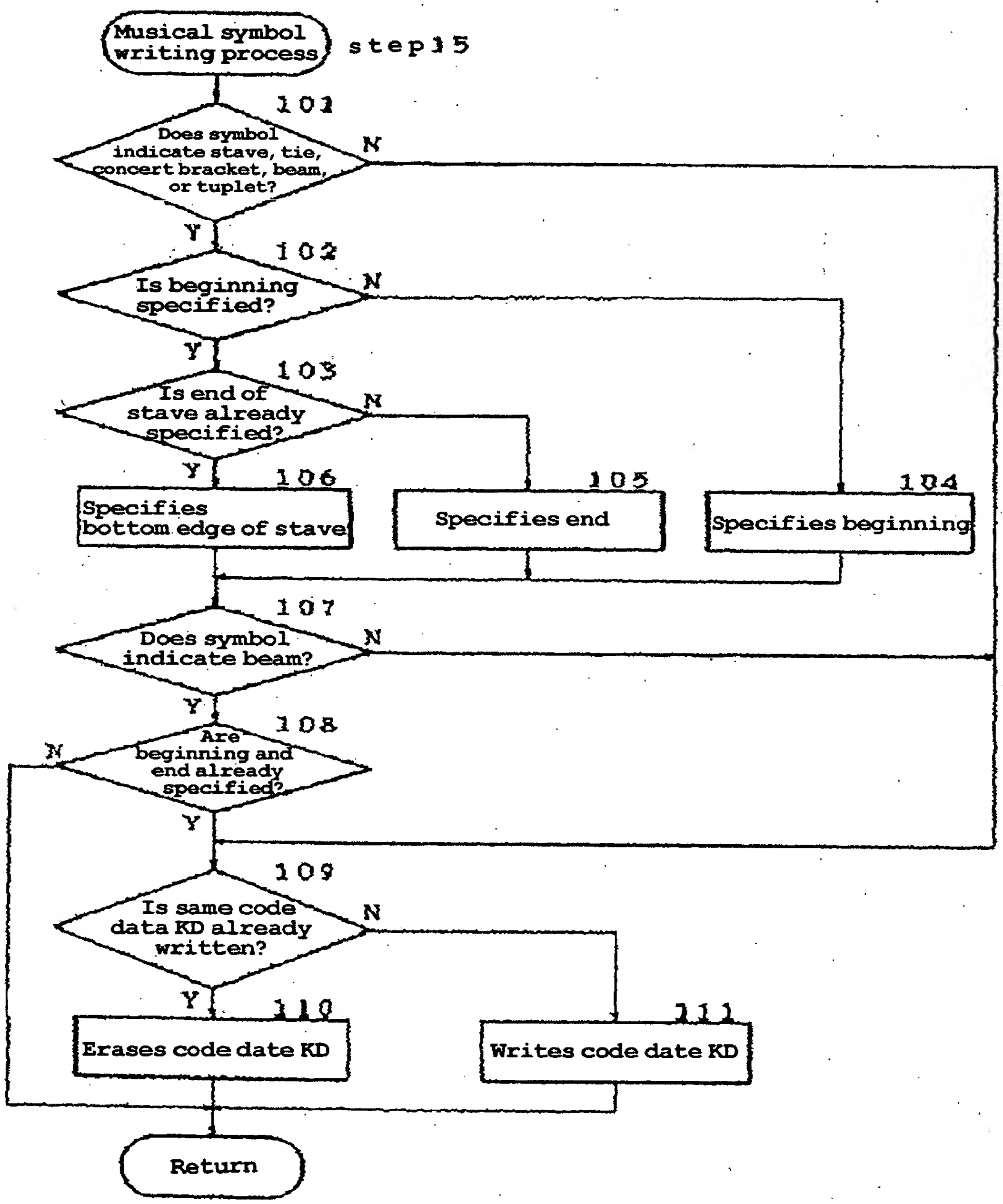

A
21
Is right mouse button clicked?
N
Y
22
Is cursor turned to musical symbol?
N
Y
23
Does symbol indicate beam?
N
Y
24
Changes flag direction
25
Cyclic change of musical symbol
26
Is center mouse button clicked?
N
Y
27
Is cursor on viewport?
N
Y
28
Displays auxiliary line
B

FIG. 9



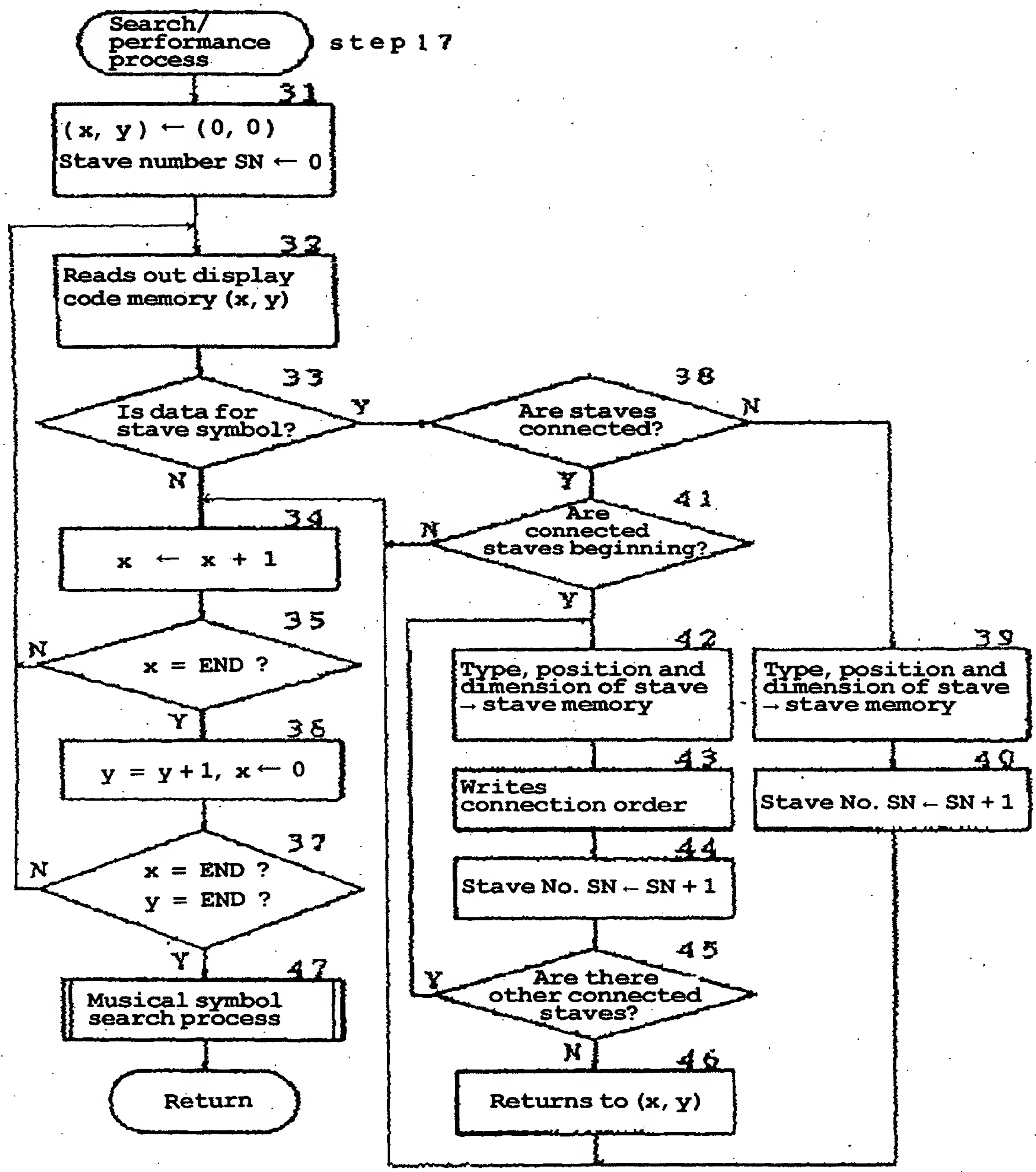
Clicks
left mouse button
Clicks
left mouse button
Beginning
End
step 104 105
Clicks left mouse button
A
B
C
D
Clicks
right mouse button
step 24
Clicks
right mouse button
Clicks
left mouse button
step 111
Fixed

FIG. 10



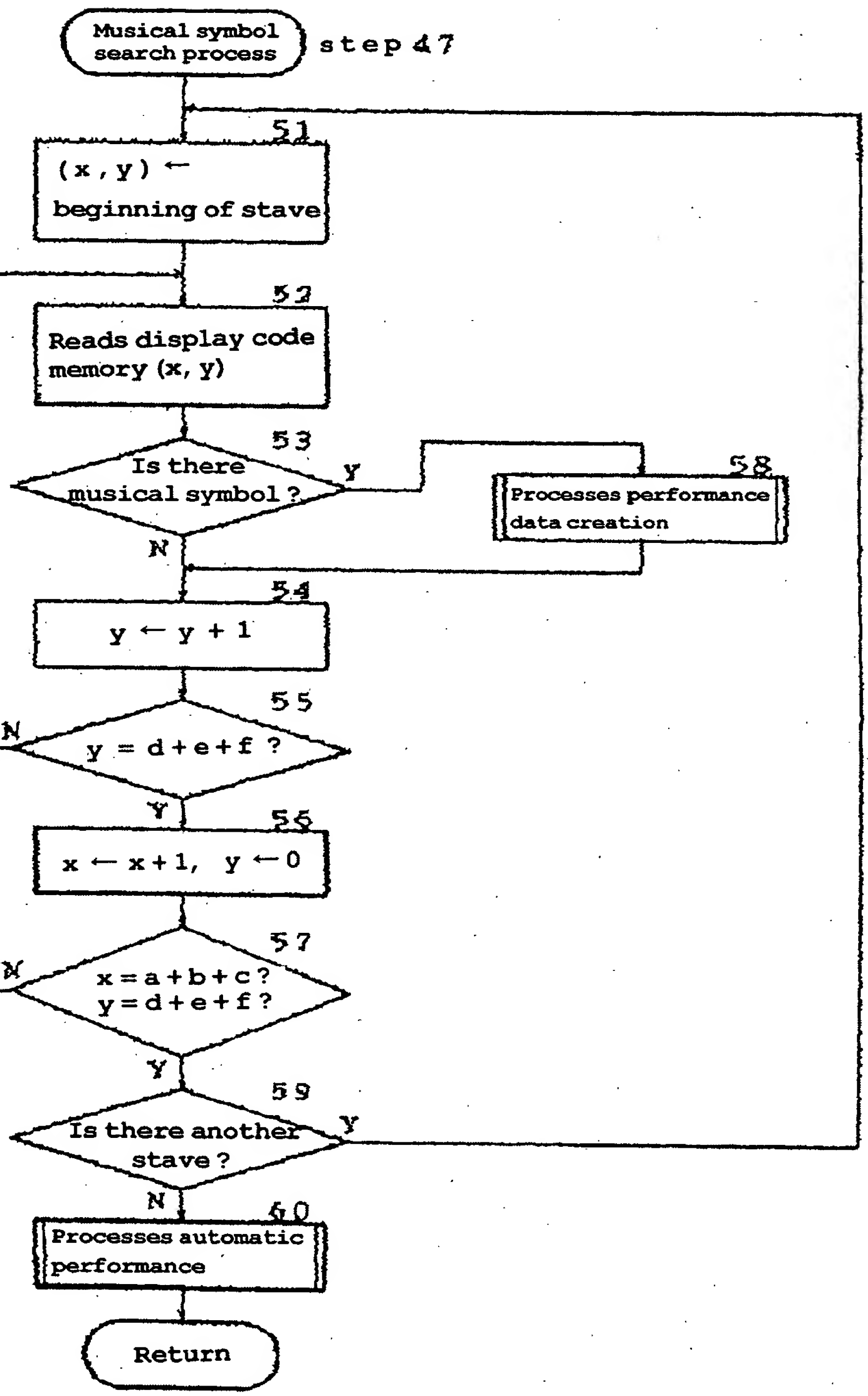
Musical symbol writing process
step 15
101
Does symbol indicate stave, tie, concert bracket, beam, or tuplet?
N
Y
102
Is beginning specified?
N
Y
103
Is end of stave already specified?
N
Y
106
Specifies bottom edge of stave
105
Specifies end
104
Specifies beginning
107
Does symbol indicate beam?
N
Y
108
Are beginning and end already specified?
N
Y
109
Is same code data KD already written?
N
Y
110
Erases code date KD
111
Writes code date KD
Return

FIG. 11



Search/ performance process
step 17
31
(x, y) ← (0, 0)
Stave number SN ← 0
32
Reads out display code memory (x, y)
33
Is data for stave symbol?
Y
N
38
Are staves connected?
N
Y
41
Are connected staves beginning?
N
Y
34
x ← x + 1
35
x = END ?
N
Y
36
y = y + 1, x ← 0
37
x = END ?
y = END ?
N
Y
47
Musical symbol search process
Return
42
Type, position and dimension of stave → stave memory
43
Writes connection order
44
Stave No. SN ← SN + 1
45
Are there other connected staves?
Y
N
46
Returns to (x, y)
39
Type, position and dimension of stave → stave memory
40
Stave No. SN ← SN + 1

FIG. 12



Musical symbol search process
step 47
51
(x, y) ←
beginning of stave
52
Reads display code memory (x, y)
53
Is there musical symbol ?
Y
58
Processes performance data creation
N
54
y ← y + 1
55
N
y = d + e + f ?
Y
56
x ← x + 1, y ← 0
57
N
x = a + b + c ?
y = d + e + f ?
Y
59
Is there another stave ?
Y
N
60
Processes automatic performance
Return

FIG. 13

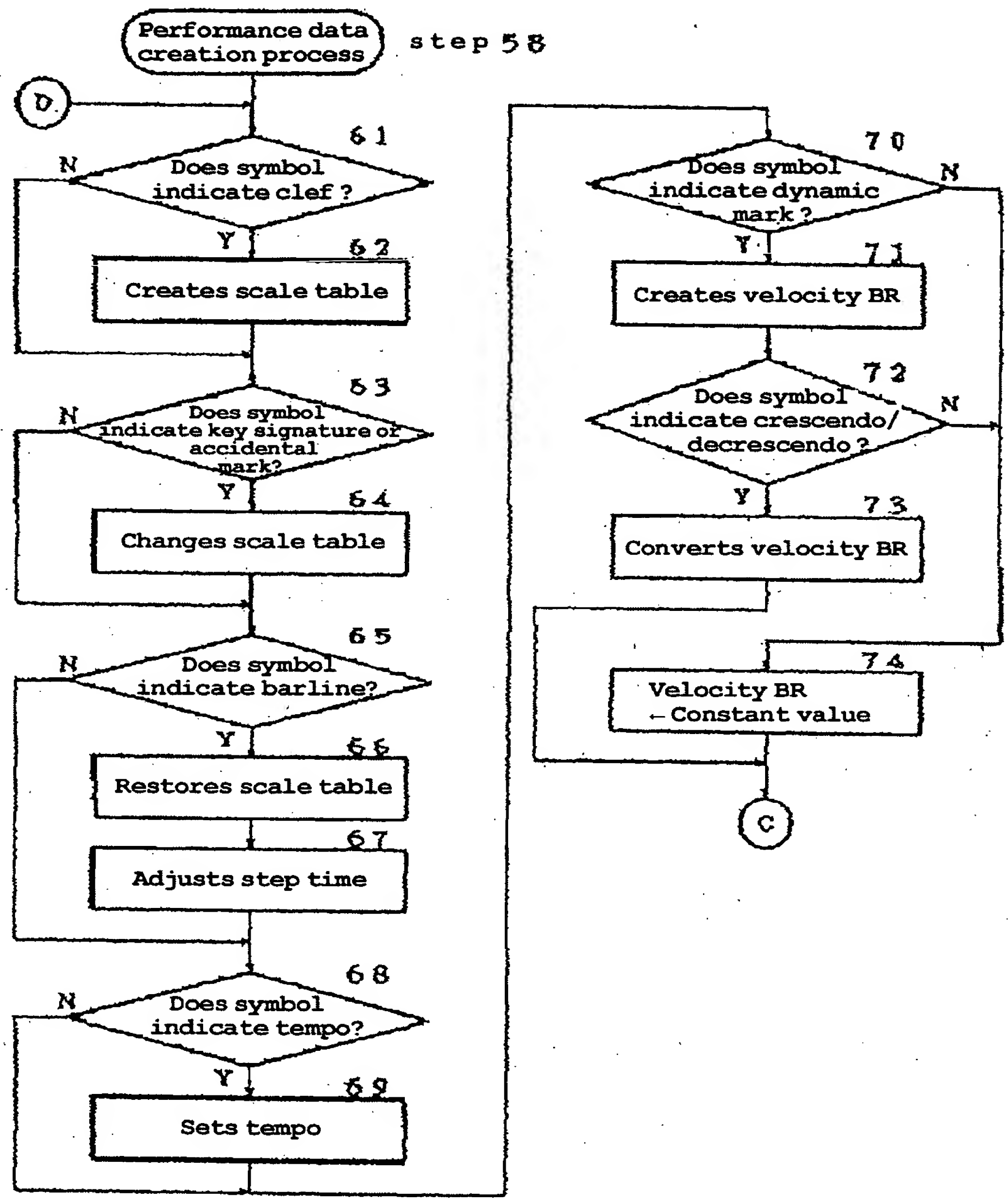
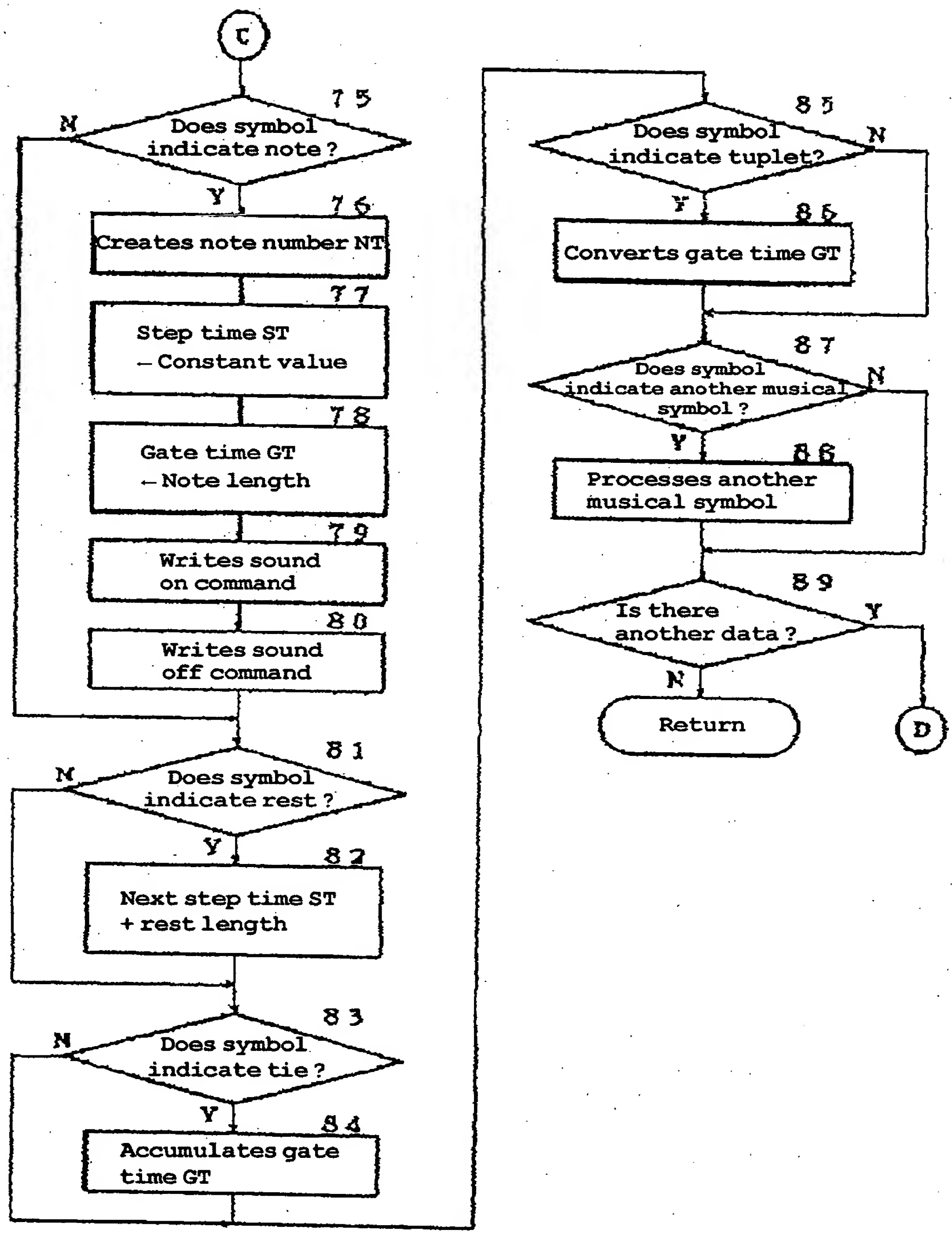

Performance data creation process
step 58
D
61
Does symbol indicate clef ?
N
Y
62
Creates scale table
63
Does symbol indicate key signature or accidental mark?
N
Y
64
Changes scale table
65
Does symbol indicate barline?
N
Y
66
Restores scale table
67
Adjusts step time
68
Does symbol indicate tempo?
N
Y
69
Sets tempo
70
Does symbol indicate dynamic mark ?
N
Y
71
Creates velocity BR
72
Does symbol indicate crescendo/ decrescendo ?
N
Y
73
Converts velocity BR
74
Velocity BR ← Constant value
C

FIG. 14



C
75
Does symbol indicate note ?
N
Y
76
Creates note number NT
77
Step time ST
← Constant value
78
Gate time GT
← Note length
79
Writes sound on command
80
Writes sound off command
81
Does symbol indicate rest ?
N
Y
82
Next step time ST
+ rest length
83
Does symbol indicate tie ?
N
Y
84
Accumulates gate time GT
85
Does symbol indicate tuplet?
N
Y
86
Converts gate time GT
87
Does symbol indicate another musical symbol ?
N
Y
88
Processes another musical symbol
89
Is there another data ?
Y
N
Return
D